IMAGE SUPER RESOLUTION

A Mini Project Report

submitted by

MUBASHIRA KM(MES20MCA-2030)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

February 2022

DECLARATION

I undersigned hereby declare that the project report **IMAGE SUPER RESOLUTION**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of Vasudevan T.V Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:

Date:

MUBASHIRA K M(MES20MCA-2030)

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **IMAGE SUPER RESOLUTION** is a bonafide record of the Mini Project work carried out by **MUBASHIRA KM(MES20MCA-2030)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

At the very outset I would like to thank the almighty's mercy towards me over the years... I wish to express my sincere thanks to my project guide Ms.Priya J.D Assistant professor, Dept. of Master of Computer Applications who guided me for the successful completeness of this project. I also thank her for valuable suggestions, guidance, constant encouragement, boundless corporation, constructive comments and motivation extended to me for completion of this project work.

I would express my sincere thanks to my internal guide Mr.Vasudevan TV, Assistant professor, Dept. of Computer Application for his immense guidance to complete the project successfully.

I would like to express my sincere thanks to all the faculty members of Master of Computer Applications department for their support and valuable suggestion for doing the project work.

Last but not least my graceful thanks to my parents, friends and also the persons who supported me directly and indirectly during the project.

MUBASHIRA KM(MES20MCA-2030)

Abstract

- High resolution (HR) image reconstruction from single low-resolution (LR) image is one of the most important vision applications. Despite numerous algorithms have been successfully proposed in recent years, efficient and robust image super resolution reconstruction is still challenging by several factors, such as inherent ambiguous mapping between the HR and LR images, necessary use exemplar images, and computational load. This paper presents the image super resolution algorithm which mainly focuses on constructing a high quality and high-resolution image from a low resolution, low quality image. It is useful in many cases like medical imaging, satellite imaging, surveillance, crime investigation and video applications etc **Keywords:**super resolution,inherent ambiguous mapping

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

[1]The central aim of super-resolution (SR) is to generate a higher resolution image from a low resolution images. The need for high resolution is common in computer vision applications for better performance in pattern recognition and analysis of images. Inspired by simple mapping functions method, a mapping matrix table of HR-LR feature patches is calculated in the training phase. Each atom of dictionary learned from LR feature patches is corresponding to a mapping matrix in the mapping matrix table. Combining this mapping table with sparse coding, high quality and HR images are reconstructed in reconstruction phase. The effectiveness and efficiency of this method is validated with experiments on the training datasets.

compared with state-of -art methods ,jagged and blurred artifacts are deppressed effectively and high reconstruction quality is acquired with less exemplar images.

1.1.1 Motivation

This paper is motivated from the advancement in machine learning algorithms for various computer vision problems. We are proposing framework that simultaneously calculates the convolutional features of low-resolution (LR) and high-resolution (HR) image patches and learns the non-linear function that maps these convolutional features of LR image patches to their corresponding HR image patches convolutional features. Here, proposed machine

1

learning based image super-resolution architecture . Super-resolution of a noisy/distorted LR images results in noisy/distorted HR images, as super-resolution process gives rise to spatial correlation in the noise, and further, it cannot be de-noised successfully.

1.2 Objective

Image super-resolution (SR) is the process of recovering high-resolution (HR) images from low-resolution (LR) images. It is an important class of image processing techniques in computer vision and image processing and enjoys a wide range of real-world applications, such as medical imaging, satellite imaging, surveillance and security, astronomical imaging, amongst others it is a process of upscaling and or improving the details within an image. Often a low resolution image is taken as an input and the same image is upscaled to a higher resolution, which is the output. The details in the high resolution output are filled in where the details are essentially unknown.

1.3 Report Organization

The project report is divided into four sections. Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

1. @ IJTSRD — Available Online @ www.ijtsrd.com ISSN No: 2456 International Research Literature Review Shalini Dubey 1 Research Scholar, Department of ECE ABSTRACT In this paper, a detailed survey study on single image super-resolution (SR) has been presented, which aims at recovering a high-resolution (HR) image from a given low-resolution (LR) one. It is always the research emphasis because of the requirement of higher definition video displaying, such as the new generation of Ultra High Definition (UHD) TVs. Super-resolution (SR) is a popular topic of image processing that focuses on the enhancement of image resolution. In general, SR takes one or several low resolution (LR) images as input and maps output images with high resolution (HR), which has been widely applied in remote sensing, medical imaging, biometric identification. Keyword: Super Resolution, Image Reconstruction, Single Image Resolution Techniques, Resolutio Enhancement. 1. INTRODUCTION Image super resolution is a image processing algorithms that produce high quality, high resolution (HR) images from a set of low quality, low resolution (LR) images or from a single image. The SR image reconstruction is useful in many practical cases wh multiple frames of the same scene can be obtained, including medical imaging, satellite imaging, and video applications [17]. The basic premise for increasing the spatial resolution in SR techniques is the availability of multiple LR images captured from the same scene. The set of source low resolution (LR) images captures only a finite amount of information from a scene; the goal of SR is to extract the independent information from each image in that set and combine the information into a single high resolution (HR) image. The requirement is of SR is that each LR image must contain some information @ IJTSRD — Available Online @ www.ijtsrd.com — Volume – 2

— Issue – 5— Jul-Aug 2018 ISSN No: 2456 - 6470 — www.ijtsrd.com — Volume International Journal of Trend in Scientific Research and Development (IJTSRD) International Open Access Journal Literature Review on Single Image Super Resolution Shalini Dubey1 , Prof. Pankaj Sahu2 , Prof. Surya Bazal2 Research Scholar, 2 Assistant Professor f ECE, GGITS, Jabalpur, Madhya Pradesh, India In this paper, a detailed survey study on single image resolution (SR) has been presented, which aims resolution (HR) image from a resolution (LR) one. It is always the research emphasis because of the requirement of gher definition video displaying, such as the new generation of Ultra High Definition (UHD) TVs. resolution (SR) is a popular topic of image on the enhancement of image takes one or several low- ion (LR) images as input and maps them as output images with high resolution (HR), which has widely applied in remote sensing, medical Super Resolution, Image Reconstruction, Single Image Resolution Techniques, Resolution Image super resolution is a image processing algorithms that produce high quality, high resolution (HR) images from a set of low quality, low resolution (LR) images or from a single image. The SR image reconstruction is useful in many practical cases where multiple frames of the same scene can be obtained, including medical imaging, satellite imaging, and The basic premise for increasing the spatial resolution in SR techniques is the availability of multiple LR images captured from the same scene. The set of source low resolution (LR) images captures only a finite amount of information from a scene; the goal of SR is to extract the independent information from each image in that set and combine the information into a single high resolution (HR) image. The requirement is of SR is that each LR image must contain some information that is unique to that image [3]. The super resolution method is to take more samples of the scene so as to get some extra information which can be used, while merging the samples to get a high resolution image. These samples can be acquired by sub changing scene, by changing the amount of blur [14] .HR means that pixel density within an image is high, and therefore an HR image can offer more de are important in many applications, The major advantage of the super resolution approach is that it may cost less and the existing LR imaging systems can be still utilized. Synthetic zooming of region of interest (ROI) is another important appli surveillance, forensic, scientific, medical, and satellite imaging [13].

Chapter 3

Methodology

3.1 Introduction

Image super resolution refers to enhancing the resolution of an image from low-resolution to high-resolution is ,an important class of image processing techniques in computer vision and image processing. Despite numerous algorithms have been successfully proposed in recent years, efficient and robust single-image super-resolution (SR) reconstruction is still challenging by several factors, such as inherent ambiguous mapping between the HR-LR images, necessary huge exemplar images, and computational load. In this paper, we proposed a new learning-based method of single-image SR. Inspired by simple mapping functions method, a mapping matrix table of HR-LR feature patches is calculated in the training phase. Each atom of dictionary learned from LR feature patches is corresponding to a mapping matrix in the mapping matrix table. Combining this mapping table with sparse coding, high quality and HR images are reconstructed in reconstruction phase. The effectiveness and efficiency of this method is validated with experiments on the training datasets. Compared with state-of-art methods, jagged and blurred artifacts are depressed effectively and high reconstruction quality is acquired with less exemplar images.

modules:

Module 1 Input Imaging, Preprocessing Image

Module 2 Data collection and preprocessing

Module 3 Comparison and Algorithm implementation

Module 4 Detection and Modifications, edge detection, replacing pixel data, writing image,

show/save

Tensorflow is a python-friendly open source library for numerical computation that make machine learning faster and easier There are six algorithms used ”sr”, ”esr”, ”dsr”, ”ddsr”, ”rnsr”, ”distilled rnsr”.

Super resolution(sr) refers to methods aiming at increasing spatial resolution of digital images.

Edge detection is a technique of image processing used to identify points in a digital image with discontinuities, simply to say, sharp changes in the image brightness. These points where the image brightness varies sharply are called the edges (or boundaries) of the image. It used to upscale low-resolution images to a higher resolution to fit the display of high-resolution monitors. The catch was that the upscaled image showed quality similar to that of rendering the image natively in a higher resolution

There are two section,main and models In main section is a run file In models,the algorithms are compared and implemented

Class creation,each class contain various algorithms which have its own method .

Super-resolution refers to the process of upscaling or improving the details of the image. ...

The original high-resolution image shows the best details when zoomed in. The other images are achieved after reconstruction after using various super-resolution methods The lower resolution input image to be upscaled The input image upscaled by nearest neighbour interpolation The input image upscaled by bi-linear interpretation, this is what your Internet browser would typically need The input image upscaled and improved by this model’s prediction The target image or ground truth, which was downscaled to create the lower resolution input. Writing image ,show and save the Image.

3.2 Developing Enviornment

Hardware specification:

i3 Processor Based Computer or higher

Memory: 1 GB RAM

Hard Drive: 50 GB

Monitor

Internet Connection

Software specification:

Language :Python

Front end : Python

Back end : python

Operating system : windows 7 or higher

IDE : PyCharm,Anaconda(spyder)

3.3 Agile Methodology

The Agile methodology is a way to manage a project by breaking it up into several phases.It involves constant collaborations with stakeholder and continues improvement at every stage .once the work begins,teams cycle through of planning ,executing and evaluating.continuous collaboration is vital ,both with team members and project stakeholders.

The project report is divided into four sections.

sprint 1

Input Imaging

Preprocessing Image

sprint 2

Data collection and preprocessing

sprint 3 Comparison and Algorithm implementation

sprint 4

Detection and Modifications

edge detection

replacing pixel data

writing image

show/save

3.4 User Story

A key component of agile software development is putting people first, and user-stories put actual end users at the center of the conversation. Stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building what they're building and what value it creates. A user story is a tool used in agile software development to capture a description of a software feature from an end user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work which drives collaboration, creativity, and a better product overall. The user story of system is given in the below table

User story ID	As a<type of user>	I want to <perform some task>	So that I can <achieve some goal>
1	User	Input imaging	Read image from a system
2	User	Preprocessing image	Resize and color scheme changes as per project requirement
3	User	Data collection and preprocessing	Cleaned final dataset
4	User	Comparison and algorithm implementation	Compare different type algorithm and implement
5	User	Edge detection	Finding boundaries Of objects within images
6	User	Replacing pixel data	Spreaded pixel data replace with neighbours
7	User	Writing image and show/save	Show and save image

Figure 3.1: user story

3.5 product backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.The product backlog is the single authoritative source for things that a team works on. That means that nothing gets done that isn't on the product backlog. Conversely, the

presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment.It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome. The Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories.The product backlog of the system is given below figure

User Story ID	Priority <High/ Medium / Low	Size(hours)	Sprint	Status/ Planned/ In progress/co mpleted	Release date	Release goal
1	medium	5	1	completed	27/12/202 1	Input imaging
2	medium	5		completed	28/12/202 1	Preprocessing image
3	medium	5	2	planned	15/01/202 2	Data collection and preprocessing
4	high	10	3	planned	23/01/202 2	Comparison and algorithm Implementation
5	high	10	4	planned	26/01/202 2	Edge detection
6	high	10		planned	05/02/202 2	Replacing pixel datas
7	medium	5		planned	12/02/202 2	Writing image and show/save

Figure 3.2: product backlog

3.6 project plan

A project plan that has a series of tasks laid out for the entire project, listing task duration, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it. Project plan is given below figure. The project has four sprints.

User	Task name	Start date	End date	Days	Status
1	Sprint 1	27/12/2021	27/12/2021	2 days	completed
2		28/12/2021	28/12/2021		completed
3	Sprint2	15/01/2022	16/01/2022	2 days	planned
4	Sprint 3	23/01/2022	24/01/2022	2 days	planned
5	Sprint4	26/01/2022	27/01/2022	6 days	planned
6		5/02/2022	6/02/2022		planned
7		12/02/2022	13/02/2022		planned

Figure 3.3: project plan

3.7 sprint plan

The sprint plan is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story. Most teams also estimate how many hours each task will take someone on the team to complete

Backlog items(user story	completion	Estimated hrs	day 1	day2	day3	day4	day5	day6	day7	day8	day9	day10	day11	day12
Input imaging	27/12/2021	5	5		0	0	0	0	0	0	0	0	0	0
Preprocessing image	28/12/2021	5	0	5	0	0	0	0	0	0	0	0	0	0
Data collection and preprocessing	15/01/2022	5	0	0	2	3	0	0	0	0	0	0	0	0
Comparison and algorithm implementation	23/01/2022	10	0	0	0	0	5	5	0	0	0	0	0	0
Edge detection	26/01/2022	10	0	0	0	0	0	0	5	5	0	0	0	0
Replacing pixel data	05/02/2022	10	0	0	0	0	0	0	0	0	5	5	0	0
Writing image Show/save	12/02/2022	5	0	0	0	0	0	0	0	0	0	0	3	2
total		50	5	5	2	3	5	5	5	5	5	5	3	2

Figure 3.4: sprint plan

3.8 sprint actual

Actual sprint backlog is what adequate sprint planning is actually done by project team there may or may not be difference in planned sprint backlog. The detailed sprint backlog (Actual) is given below

Backlog items(user story	completion	Estimated hrs	day 1	day2	day3	day4	day5	day6	day7	day8	day9	day10	day11
Input imaging	27/12/2021	5	5		0	0	0	0	0	0	0	0	0
Preprocessing image	28/12/2021	5	0	5	0	0	0	0	0	0	0	0	0
Data collection and preprocessing	15/01/2022	5	0	0	2	3	0	0	0	0	0	0	0
Comparison and algorithm implementation	23/01/2022	10	0	0	0	0	5	5	0	0	0	0	0
Edge detection	26/01/2022	0	0	0	0	0	0	0	5	5	0	0	0
Replacing pixel Data	5/02/2022	0	0	0	0	0	0	0	0	0	5	5	0
Writing image Show/save	12/02/2022	0	0	0	0	0	0	0	0	0	0	3	2
total		50	5	5	2	3	5	5	5	5	5	8	2

Figure 3.5: sprint actual

Chapter 4

Results and Discussions

4.1 Datasets

DIV2K is a popular single-image super-resolution dataset which contains 1,000 images with different scenes and is splitted to 800 for training, 100 for validation and 100 for testing. ... This dataset contains low resolution images with different types of degradations.It was collected for NTIRE2017 and NTIRE2018 Super-Resolution Challenges in order to encourage research on image super-resolution with more realistic degradation. This dataset contains low resolution images with different types of degradations. Apart from the standard bicubic downsampling, several types of degradations are considered in synthesizing low resolution images for different tracks of the challenges. Track 2 of NTIRE 2017 contains low resolution images with unknown x4 downscaling. Track 2 and track 4 of NTIRE 2018 correspond to realistic mild ×4 and realistic wild ×4 adverse conditions, respectively. Low-resolution images under realistic mild x4 setting suffer from motion blur, Poisson noise and pixel shifting. Degradations under realistic wild x4 setting are further extended to be of different levels from image to image.

4.2 Results

image super resolution using machine learning results, various algorithms that combines and implements can generate better photo realistic super resolved images.

Figure 4.1: t2

Figure 4.2: t2$_{i}ntermediate$

Figure 4.3: t2$_{s}caled$

Chapter 5

Conclusions

In this paper, we have given an extensive survey on recent advances in image super-resolution with machine learning. We mainly discussed the improvement of supervised and unsupervised SR, and also introduced some domain-specific applications. Despite great success, there are still many unsolved problems. Thus in this section, we will point out these problems explicitly and introduce some promising trends for future evolution. We hope that this survey not only provides a better understanding of image SR for researchers but also facilitates future research activities and application developments in this field

References

[1]. H. Greenspan, “Super-resolution in medical imaging,” The Computer Journal, vol. 52, 2008.

[2]. J. S. Isaac and R. Kulkarni, “Super resolution techniques for medical image processing,” in ICTSD, 2015.

[3]. Y. Huang, L. Shao, and A. F. Frangi, “Simultaneous superresolution and cross-modality synthesis of 3d medical images using weakly-supervised joint convolutional sparse coding,” in CVPR, 2017.

[4]. L. Zhang, H. Zhang, H. Shen, and P. Li, “A super-resolution reconstruction algorithm for surveillance images,” Elsevier Signal Processing, vol. 90, 2010.

[5]. P. Rasti, T. Uiboupin, S. Escalera, and G. Anbarjafari, “Convolutional neural network super resolution for face recognition in surveillance monitoring,” in AMDO, 2016.

[6]. D. Dai, Y. Wang, Y. Chen, and L. Van Gool, “Is image superresolution helpful for other vision tasks?” in WACV, 2016.

[7]. M. Haris, G. Shakhnarovich, and N. Ukita, “Task-driven super resolution: Object detection in low-resolution images,” Arxiv:1803.11316, 2018.

[8]. M. S. Sajjadi, B. Scholkopf, and M. Hirsch, “Enhancenet: Single ¨ image super-resolution through automated texture synthesis,” in ICCV, 2017.

Appendix

Source Code

main.py

```
import models
import argparse
import tensorflow as tf

parser = argparse.ArgumentParser(description="Up-Scales an image using Image Super Resolution Model")
#parser.add_argument("imgpath", type=str, nargs="+", help="Path to input image")
parser.add_argument("--model", type=str, default="distilled_rnsr", help="Use either image super resolution (sr), ""expanded
      super resolution (esr), denoising auto encoder sr (dsr), ""deep denoising sr (ddsr) or res net sr (rnsr)")
parser.add_argument("--scale", default=2, help='Scaling factor. Default = 2x')
parser.add_argument("--mode", default="patch", type=str, help='Mode of operation. Choices are "fast" or "patch"')
parser.add_argument("--save_intermediate", dest='save', default='True', type=str,
                  help="Whether to save bilinear upscaled image")
parser.add_argument("--suffix", default="scaled", type=str, help='Suffix of saved image')
parser.add_argument("--patch_size", type=int, default=8, help='Patch Size')

def strToBool(v):
   return v.lower() in ("true", "yes", "t", "1")

args = parser.parse_args()


suffix = args.suffix

model_type = str(args.model).lower()
if not model_type in ["sr", "esr", "dsr", "ddsr", "rnsr", "distilled_rnsr"]:
   raise ValueError('Model type must be either "sr", "esr", "dsr", '
                '"ddsr", "rnsr" or "distilled_rnsr"')

mode = str(args.mode).lower()
assert mode in ['fast', 'patch'], 'Mode of operation must be either "fast" or "patch"'

scale_factor = int(args.scale)
save = strToBool(args.save)

patch_size = int(args.patch_size)
assert patch_size > 0, "Patch size must be a positive integer"

with tf.device('/CPU:0'):
 #path = args.imgpath
   path = 'input_images/images.bmp'
   for p in path:
```

```
    if model_type == "sr":
        model = models.ImageSuperResolutionModel(scale_factor)
    elif model_type == "esr":
        model = models.ExpantionSuperResolution(scale_factor)
    elif model_type == "dsr":
        model = models.DenoisingAutoEncoderSR(scale_factor)
    elif model_type == "ddsr":
        model = models.DeepDenoiseSR(scale_factor)
    elif model_type == "rnsr":
        model = models.ResNetSR(scale_factor)
    elif model_type == "distilled_rnsr":
        model = models.DistilledResNetSR(scale_factor)
    else:
        model = models.DistilledResNetSR(scale_factor)

    model.upscale(path, save_intermediate=save, mode=mode, patch_size=patch_size,
17
 suffix=suffix)
```